--- Nebenläufigkeit (Concurrency Control)

Prof. Dr. Ingo Claßen

Mittwoch, 16. Februar 2022 10:12

- Anforderungen an Transaktionen
- Anomalien
- Mechanismen
- Isolationsstufen
- Snapshot Isolation

Anforderungen an Transaktionen (ACID)

Prof. Dr. Ingo Claßen

Dienstag, 8. Februar 2022 09:32

Atomarität (Atomicity)
- Alles oder Nichts
- Alle Aktionen einer Transaktion werden bestätigt oder keine

Konsistenz (Consistency)
- Eine Transaktion kann nur bestätigt werden, wenn keine Konsistenbedingungen verletzt sind
- Ist eigentlich ein Konzept der Anwendung

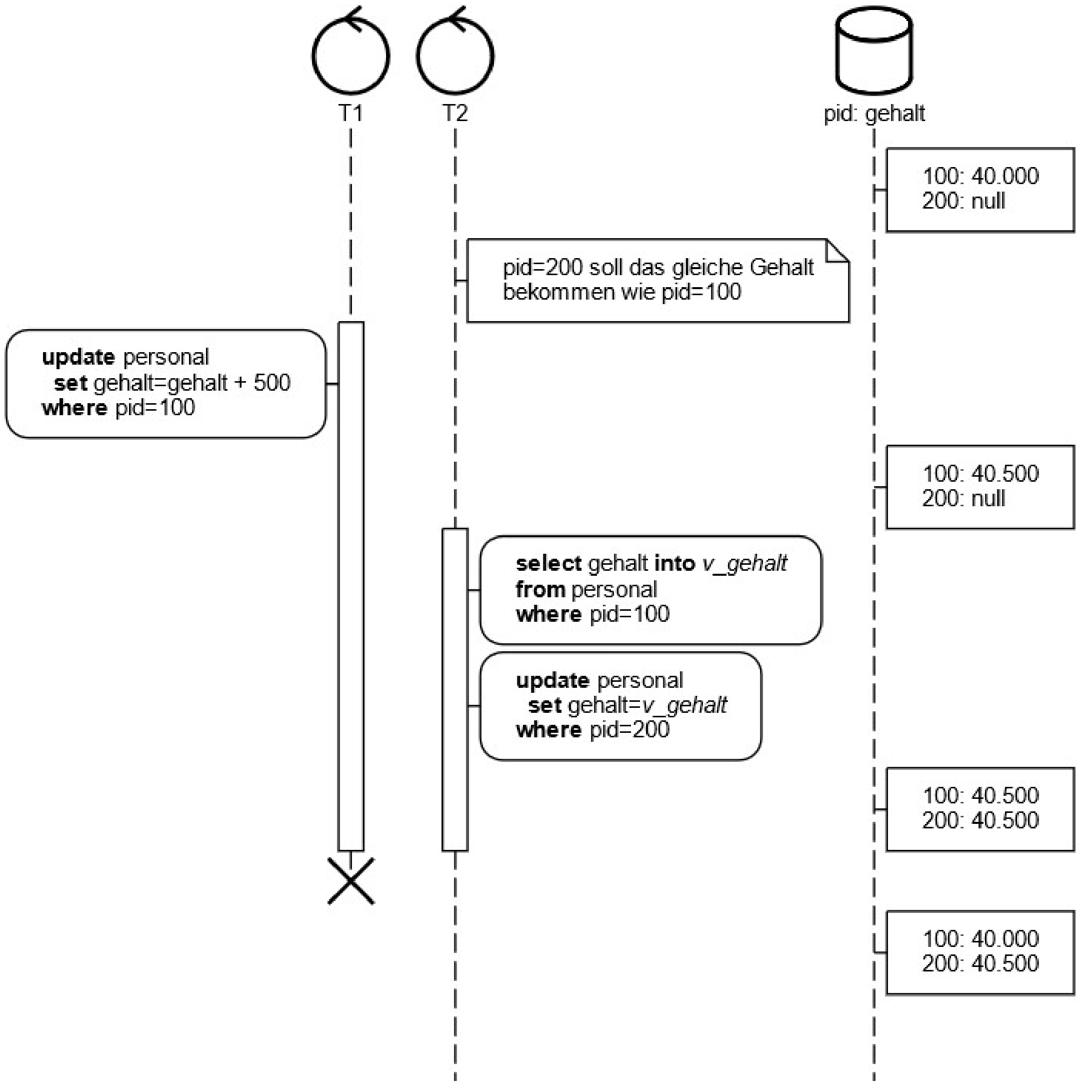
Isolation (Isolation)
- Nebenläufige/Parallele Transktionen beinflussen sich nicht auf unerlaubte Art und Weise.
- Es gibt verschiedene Stufen der Isolation, deren stärkste Serialisierbarket ist
- Stärkere Stufe bedeutet geringere Leistung

Dauerhaftigkeit (Durability)
- Veränderungen bestätigter Transkationen haben für Folgetransaktionen Bestand
- Solange bis sie durch weitere Transaktionen bestätigt verändert werden
- Auch im Falle von Fehler (z.B. Festplatte defekt)

Anomalie - Dirty Read

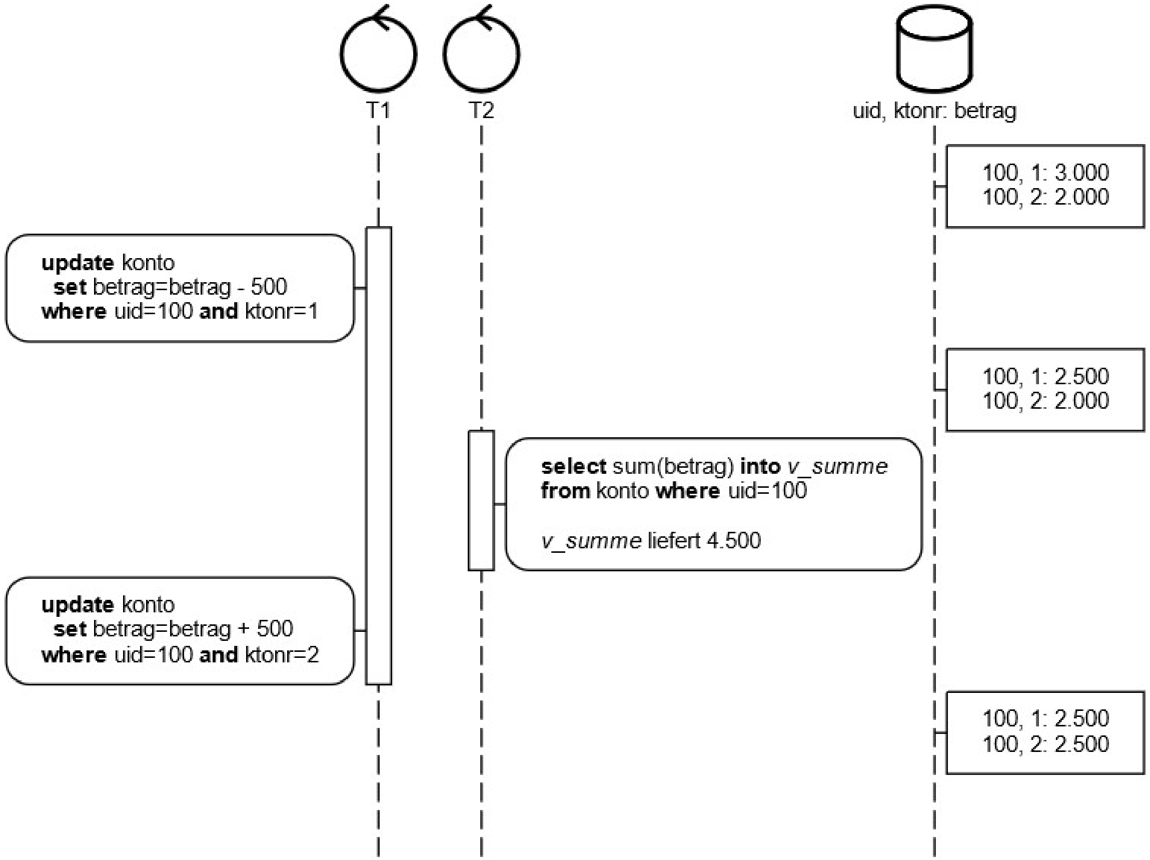
Prof. Dr. Ingo Claßen

Dienstag, 15. Februar 2022 15:18



Anomalie - Read Skew 1

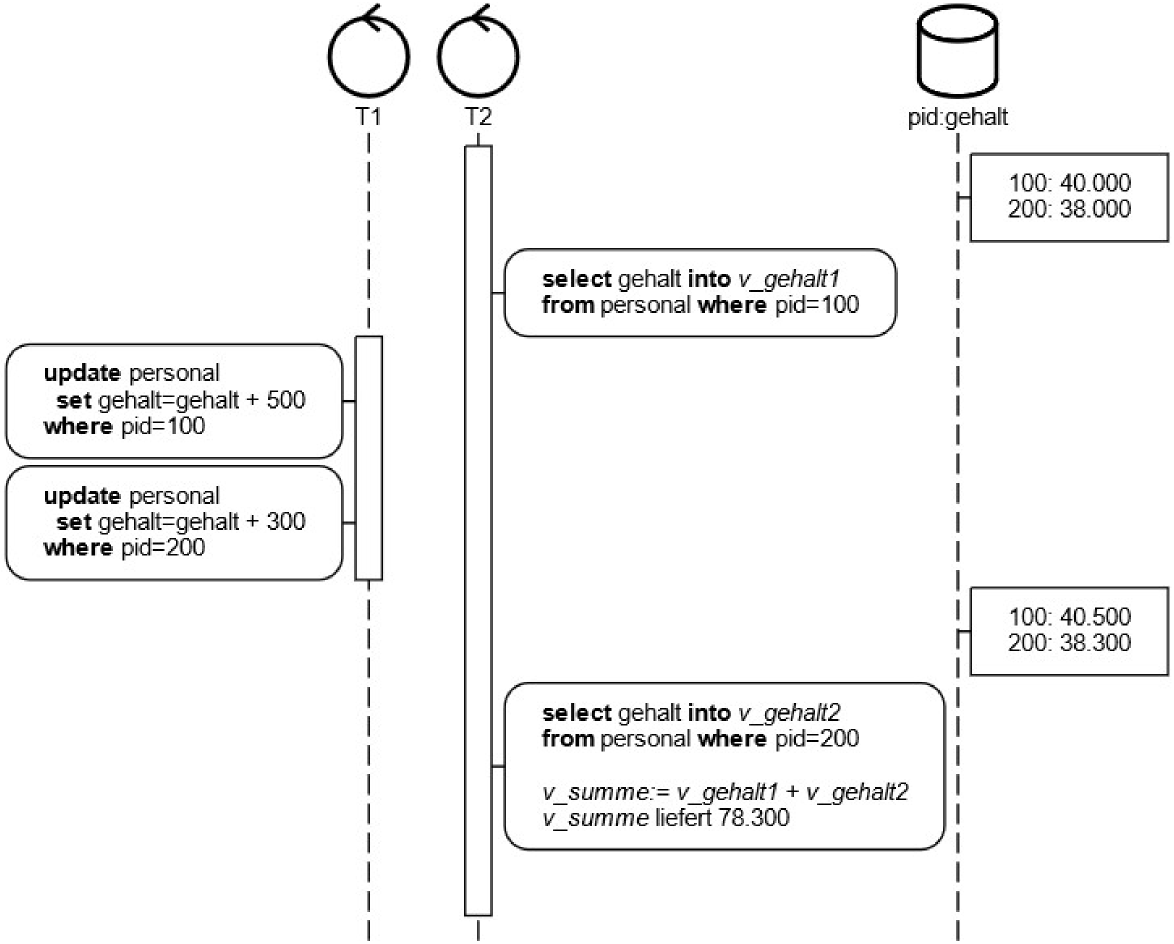
Dienstag, 15. Februar 2022 15:18



Anomalie - Read Skew 2

Prof. Dr. Ingo Claßen

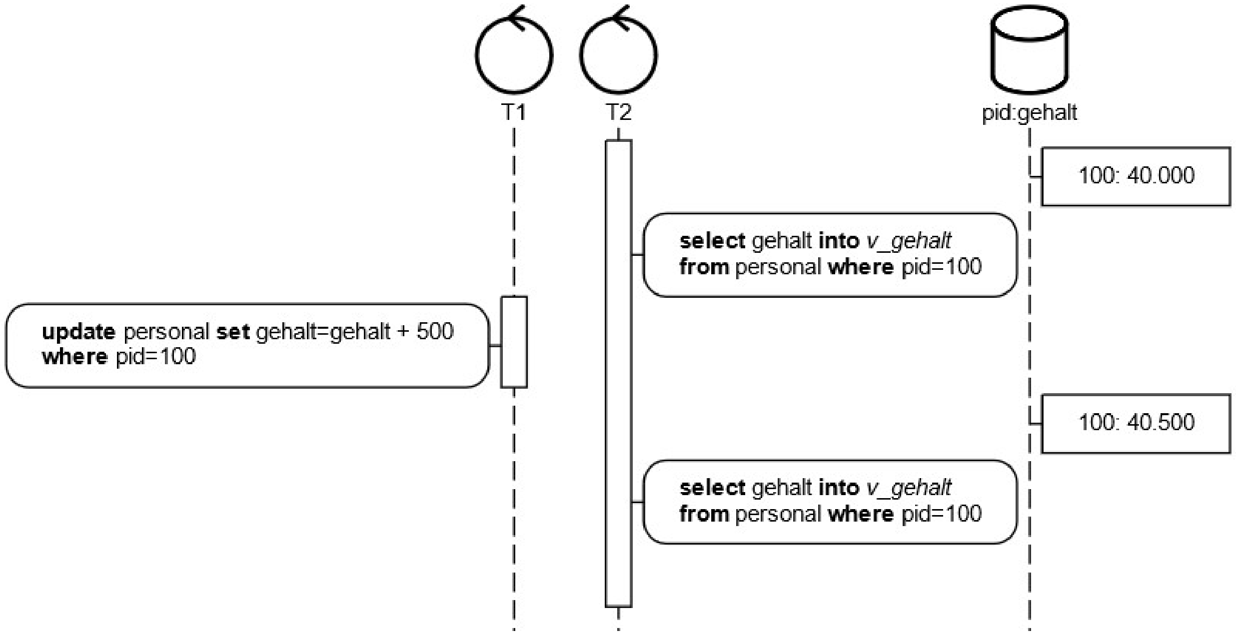
Dienstag, 15. Februar 2022 15:18



Anomalie - Non Repeatable Read

Prof. Dr. Ingo Claßen

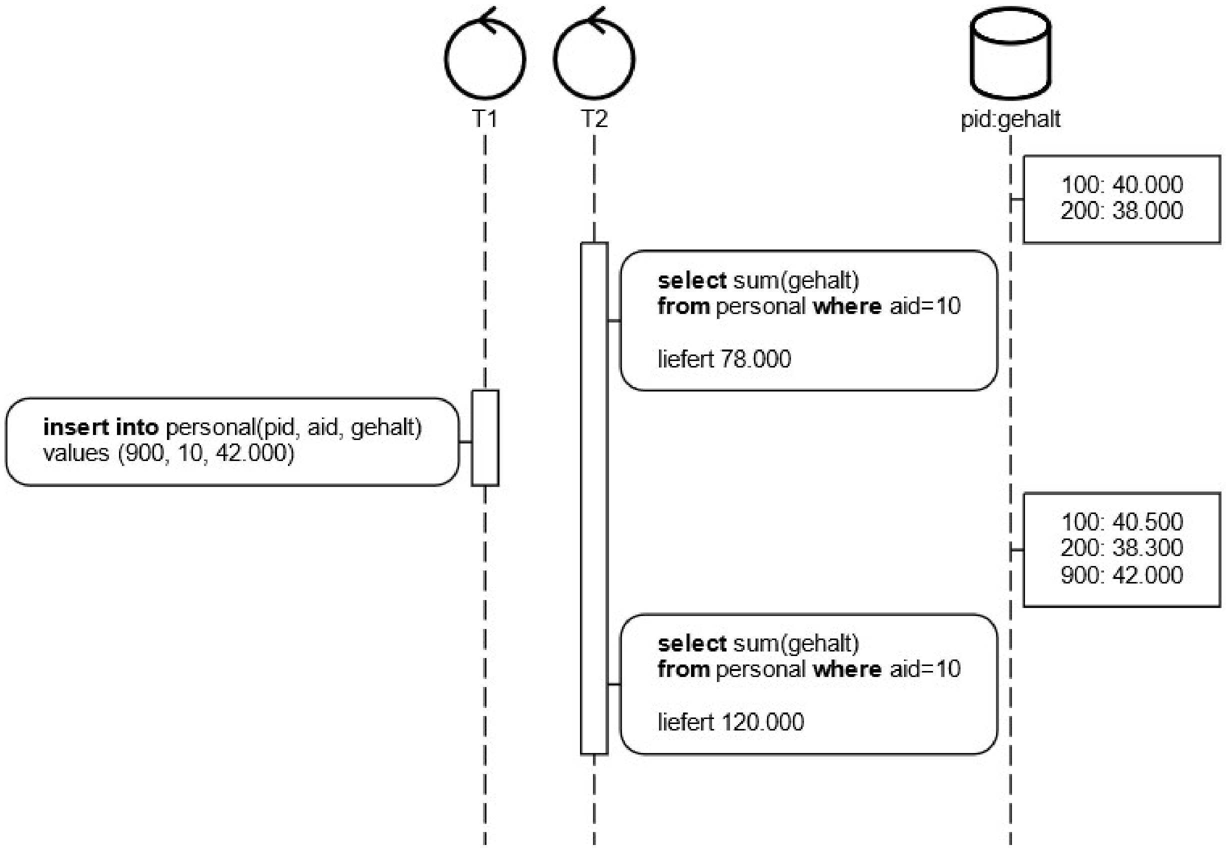
Dienstag, 15. Februar 2022 15:18



Anomalie - Phantom

Prof. Dr. Ingo Claßen

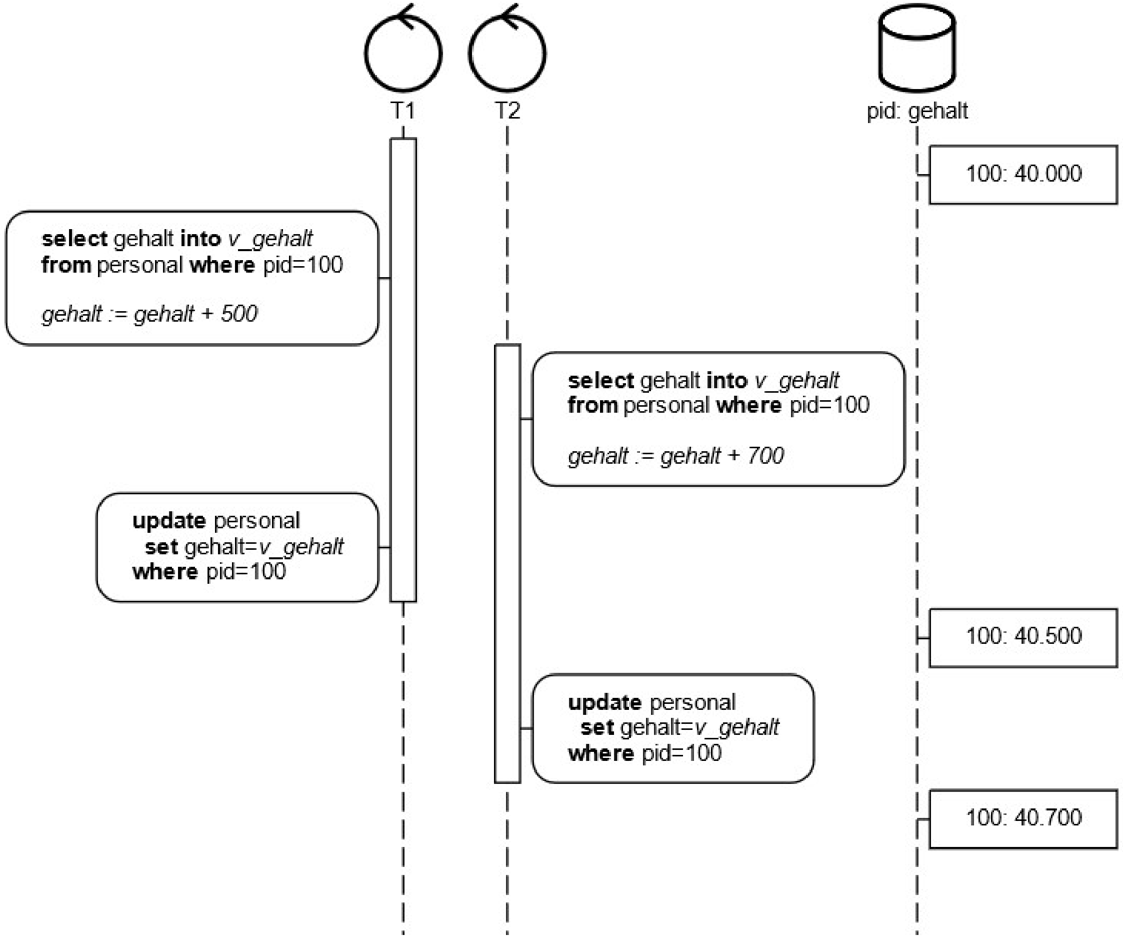
Dienstag, 15. Februar 2022 15:18



Anomalie - Lost Update

Prof. Dr. Ingo Claßen

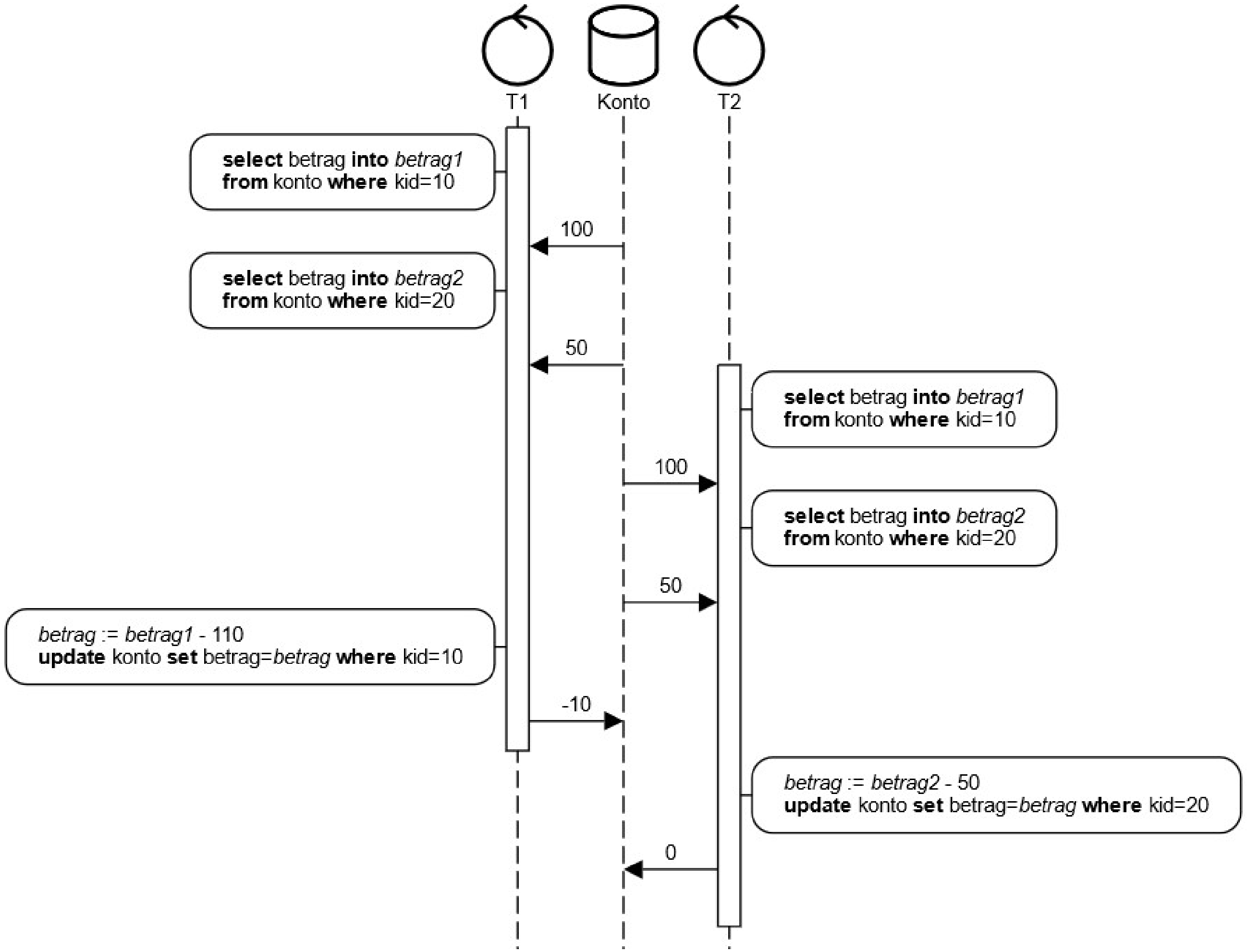
Dienstag, 15. Februar 2022 15:18



Anomalie - Write Skew

Prof. Dr. Ingo Claßen

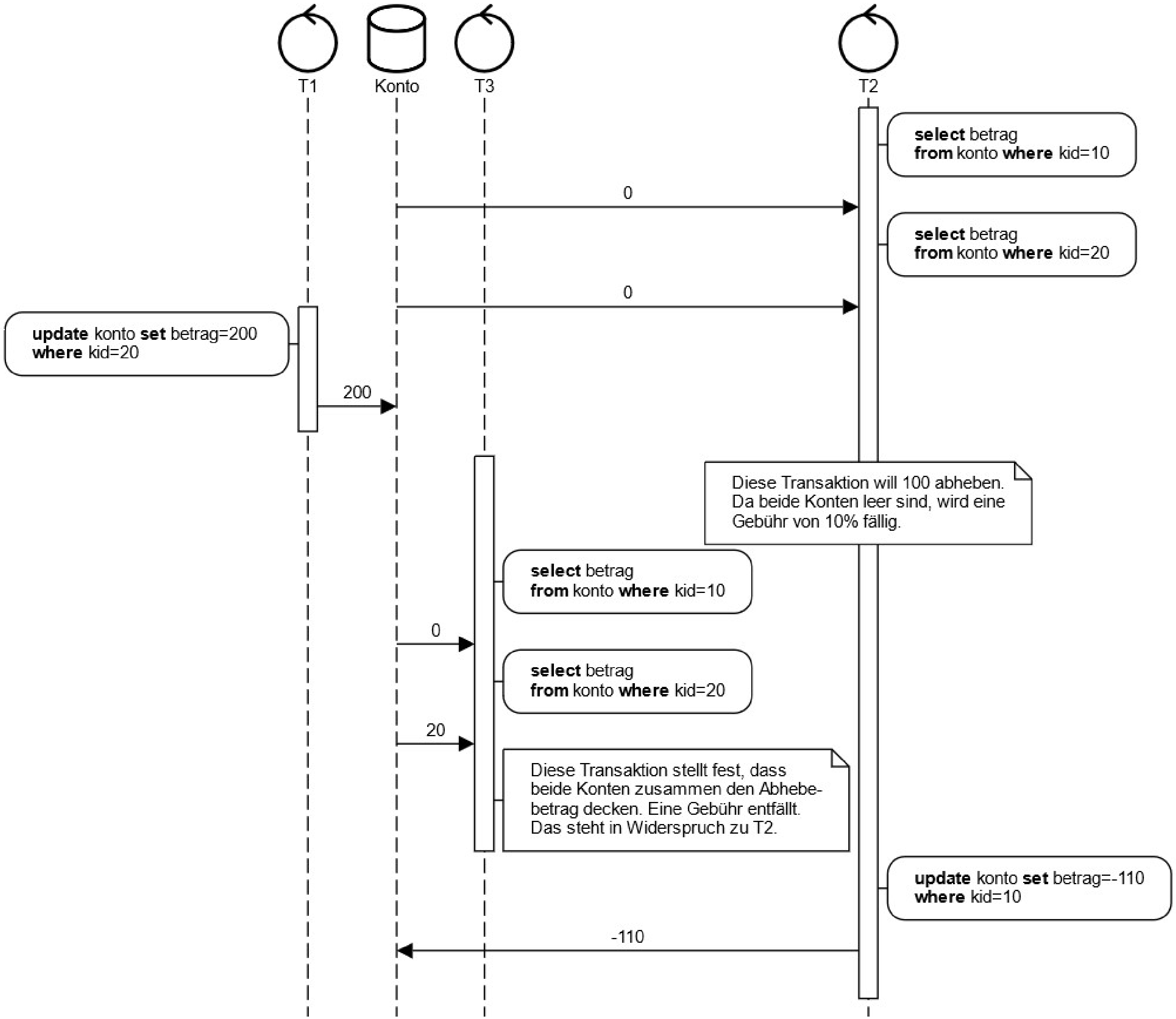
Dienstag, 15. Februar 2022 15:18



Anomalie - Read Only Anomaly

Prof. Dr. Ingo Claßen

Dienstag, 15. Februar 2022 15:18



Strikte 2-Phasen-Sperren (Strict two-phase locking , 2PL)

- Lesesperren (Shared Locks , Slocks) vor jedem Lesen
- Schreibsperren (Exclusive Locks, Xlocks) vor jedem Schreiben
- Sperren auf existierende Datensätze (Row Level Locks)
- Sperren basierend auf Prädikate (Predicate Locks) zur Vehinderung von Phantomen
- Sperren werden bis zum Transkationsende gehalten

Optimistic Concurrency Control (OCC)

- Lesen und Schreiben ohne Sperren (auf Kopien)
- Historie der Lese- und Schreibvorgänge
- Überprüfung auf Konflikte vor Commit
- Abbruch bei Konflikt

Multi-Version Concurrency Control (MVCC)

- Keine Lesesperren
- Änderungen erzeugen neue Versionen
- Transaktionen sehen Version der letzten bestätigten Änderung zu Transaktionsbeginn
- Abbruch bei Konflikt

Optimistische Verfahren auf Anwendungsebene

- Explizite Versionsfelder
- Änderung nur wenn keine Versionsänderung in Datenbank
- Bsp. OR-Mapper

Isolationsstufen

Prof. Dr. Ingo Claßen

Dienstag, 8. Februar 2022 10:39

SQL-Isolationsstufen

Read Uncommitted
- Lesen nichtbestätigter Änderungen möglich

Read Committed
- Lesen nur von bestätigten Änderungen möglich

Repetable Read
- Wiederholtes Lesen in der Transaktion liefert gleichen Wert

Serializable
- Nebenläufige Ausführung von Transkationen muss einer serialisierten Ausführung entsprechen

Produktspezifische Isolationsstufen

Cursor Stability (DB2, ..)
- Basiert auf Datenbank-Cursor
- Sperre bleibt solange bestehen, bis auf den nächsten Datensatz im Cursor weitergeschaltet wird

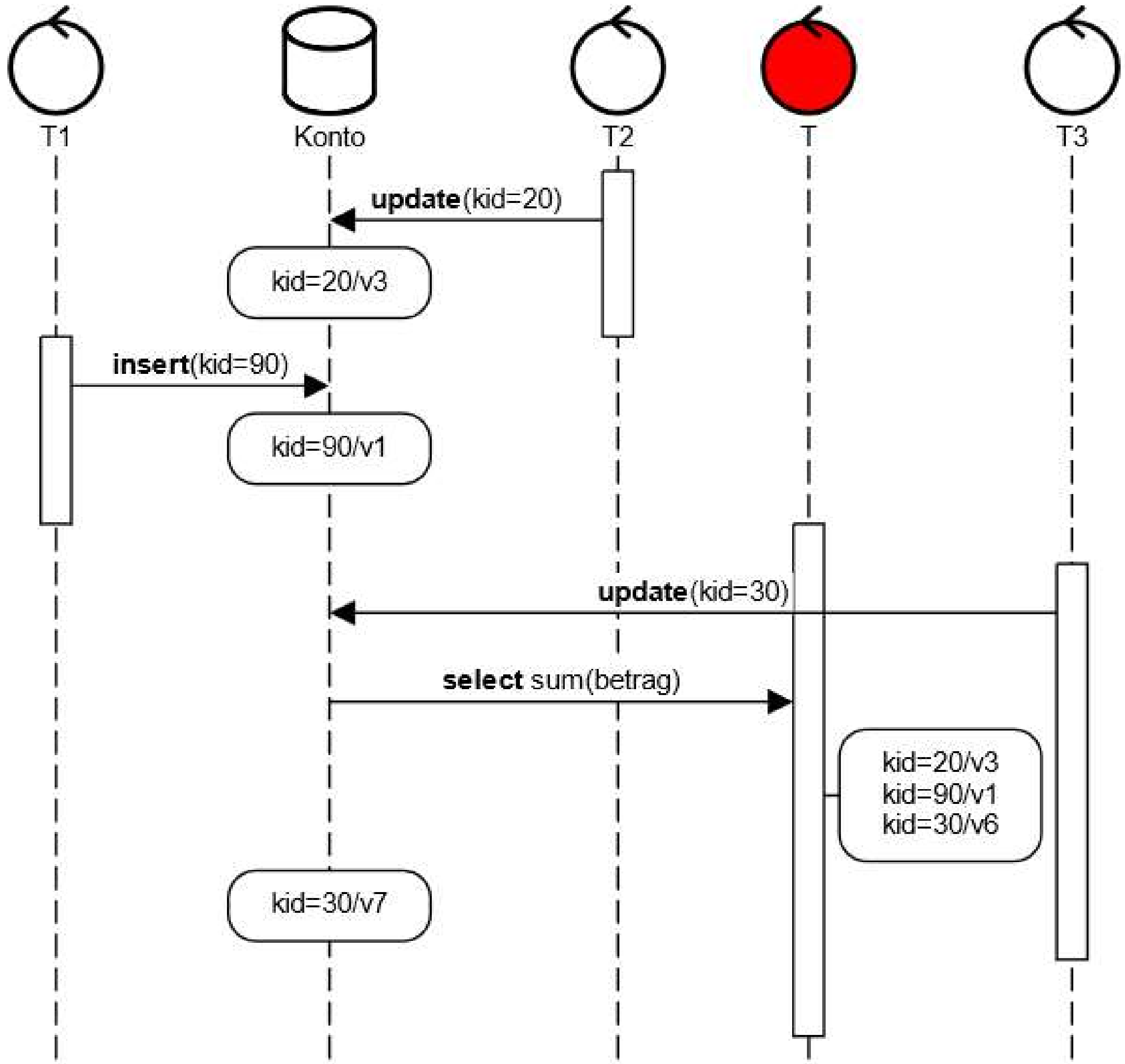
Snapshot Isolation (PostgreSQL, …)
- Basiert auf MVCC
- First Committer wins

Read Consistency (Oracle)
- Basiert auf MVC
- First Updater wins

Snapshot Isolation

Dienstag, 15. Februar 2022 15:18

Prof. Dr. Ingo Claßen



SI-Anomalien

Mittwoch, 16. Februar 2022 10:52

Prof. Dr. Ingo Claßen

Verhinderte Anomalien

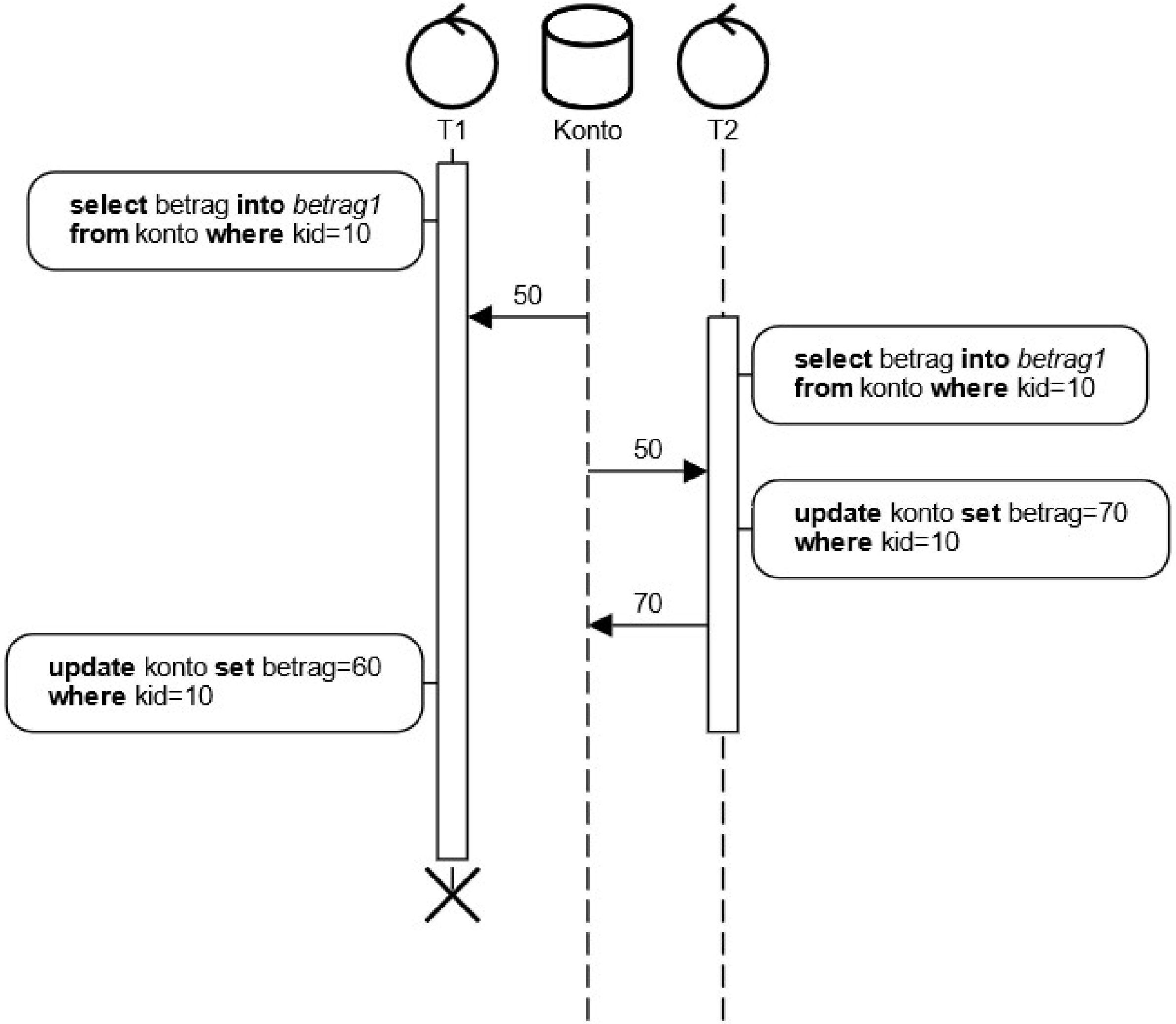
- Dirty Read
- Read Skew
- Non Repeatable Read
- Lost Update

Mögliche Anomalien

- Write Skew
- Read Only Anomaly

SI - No Lost Update 1

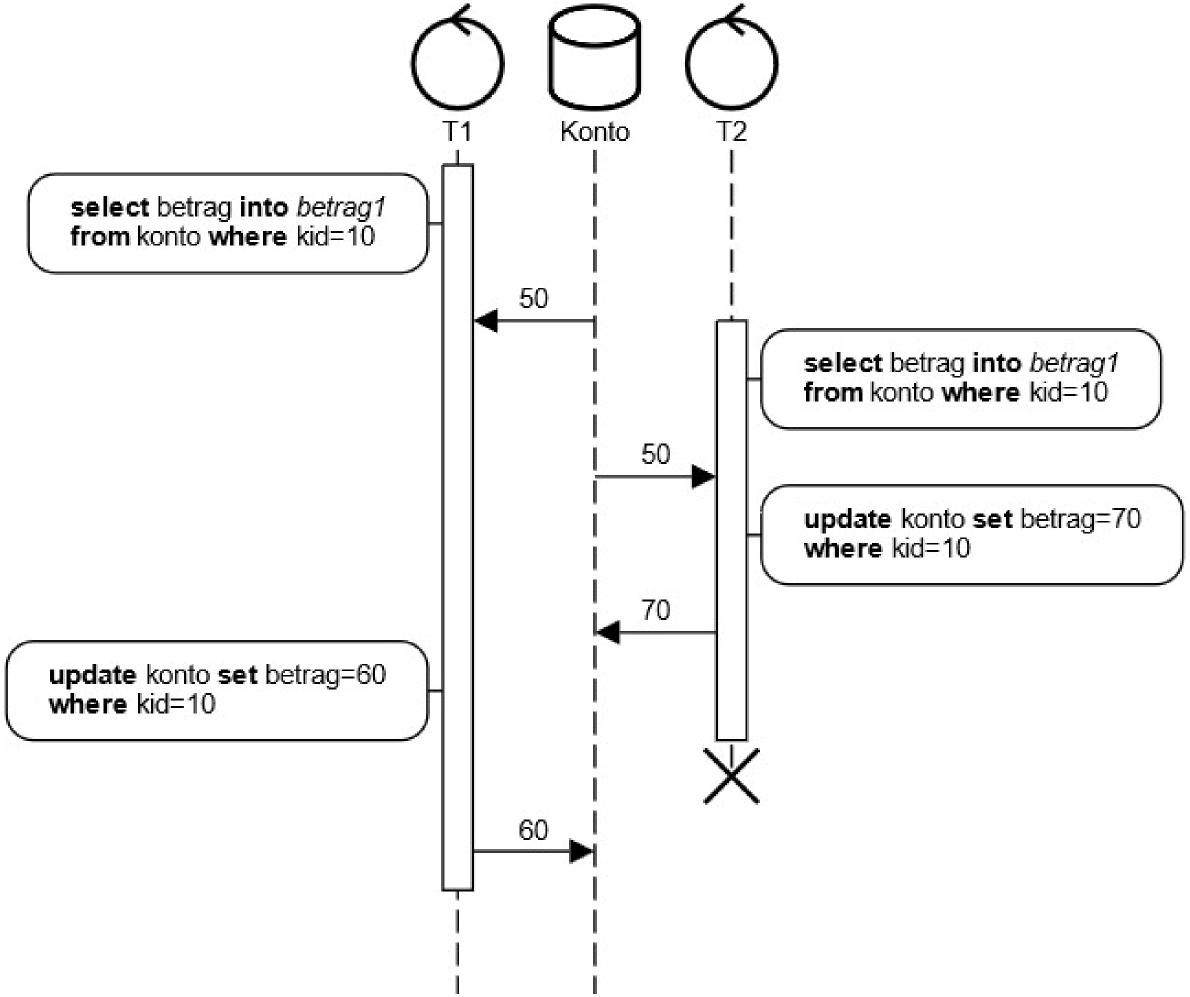
Dienstag, 15. Februar 2022 15:18



SI - No Lost Update 2

Prof. Dr. Ingo Claßen

Dienstag, 15. Februar 2022 15:18



--- Beispiele Postgres

Prof. Dr. Ingo Claßen

Montag, 28. Februar 2022 15:49

Montag, 28. Februar 2022 15:50

```
k |v   |
--+----+
10|10-a|
```

<table>
<tr><td>kv</td><td>10|10-a|</td><td>T1</td><td>T2</td><td>T3</td></tr>
<tr><td>T2</td><td>begin transaction isolation level read committed;</td><td></td><td></td><td></td></tr>
<tr><td>T3</td><td>begin transaction isolation level repeatable read;</td><td></td><td></td><td></td></tr>
<tr><td>T1</td><td>begin transaction isolation level read committed;
update kv set v = '10-b' where k=10;
select * from kv;
commit;</td><td>10|10-b|</td><td></td><td></td></tr>
<tr><td>T2</td><td>select * from kv;
commit;</td><td></td><td>10|10-b|</td><td></td></tr>
<tr><td>T3</td><td>select * from kv;
commit;</td><td></td><td></td><td>10|10-a|</td></tr>
</table>

2

Montag, 28. Februar 2022 15:50

<table>
<tr><td>kv</td><td>10|10-a|</td><td>T1</td><td>T2</td></tr>
<tr><td>T1</td><td>begin transaction isolation level serializable;
update kv set v = '10-b' where k=10;
select * from kv;</td><td>10|10-b|</td><td></td></tr>
<tr><td>T2</td><td>begin transaction isolation level serializable;
update kv set v = '10-c' where k=10;</td><td></td><td>wartet</td></tr>
<tr><td>T1</td><td>commit;</td><td></td><td>Serialisierungsfehler</td></tr>
<tr><td>T2</td><td>rollback;</td><td></td><td></td></tr>
<tr><td>kv</td><td>10|10-b|</td><td></td><td></td></tr>
</table>

kv	10|10-a|	T1	T2
T1	begin transaction isolation level serializable; update kv set v = '10-b' where k=10; select * from kv;	10|10-b|	
T2	begin transaction isolation level serializable; update kv set v = '10-c' where k=10;		wartet
T1	rollback;		
T2	select * from kv; commit;		läuft weiter 10|10-c|
kv	10|10-c|		

Montag, 28. Februar 2022 15:50

kv	10|10-a| 20|20-a|	T1	T2
T1	begin transaction isolation level serializable; update kv set v = '10-b' where k=10; select * from kv order by k;	10|10-b| 20|20-a|	
T2	begin transaction isolation level serializable; update kv set v = '20-b' where k=20; select * from kv order by k;		10|10-a| 20|20-b|
T1	commit;		Serialisierungsfehler
T2	rollback;		
kv	10|10-b| 20|20-a|		

Prof. Dr. Ingo Claßen

Montag, 28. Februar 2022 15:50

kv	10|10-a| 20|20-a|	T1	T2
T1	begin transaction isolation level serializable; update kv set v = '10-b' where k=10; select * from kv order by k;	10|10-b| 20|20-a|	
T2	begin transaction isolation level serializable; update kv set v = '20-b' where k=20; select * from kv order by k;		10|10-a| 20|20-b|
T1	commit;		
T2	commit;		
kv	10|10-b| 20|20-b|		

6

Montag, 28. Februar 2022 15:50

<table>
<tr><td>kv</td><td>10|10-a|
20|20-a|</td><td>T1</td><td>T2</td><td>T3</td></tr>
<tr><td>T2</td><td>begin transaction isolation level serializable;
select * from kv order by k;</td><td></td><td>10|10-a|
20|20-a|</td><td></td></tr>
<tr><td>T1</td><td>begin transaction isolation level serializable;
update kv set v = '10-b' where k=10;
select * from kv order by k;
commit;</td><td>10|10-b|
20|20-a|</td><td></td><td></td></tr>
<tr><td>T3</td><td>begin transaction isolation level serializable;
select * from kv order by k;
commit;</td><td></td><td></td><td>10|10-b|
20|20-a|</td></tr>
<tr><td>T2</td><td>update kv set v = '20-b' where k=20;</td><td></td><td>Serialisierungsfehler</td><td></td></tr>
<tr><td>T2</td><td>rollback;</td><td></td><td></td><td></td></tr>
<tr><td>kv</td><td>10|10-b|
20|20-a|</td><td></td><td></td><td></td></tr>
</table>

7

Montag, 28. Februar 2022 15:50

<table>
<tr><td>kv</td><td>10|10-a|
20|20-a|</td><td>T1</td><td>T2</td><td>T3</td></tr>
<tr><td>T2</td><td>begin transaction isolation level repeatable read;
select * from kv order by k;</td><td></td><td>10|10-a|
20|20-a|</td><td></td></tr>
<tr><td>T1</td><td>begin transaction isolation level repeatable read;
update kv set v = '10-b' where k=10;
select * from kv order by k;
commit;</td><td>10|10-b|
20|20-a|</td><td></td><td></td></tr>
<tr><td>T3</td><td>begin transaction isolation level repeatable read;
select * from kv order by k;
commit;</td><td></td><td></td><td>10|10-b|
20|20-a|</td></tr>
<tr><td>T2</td><td>update kv set v = '20-b' where k=20;
select * from kv order by k;</td><td></td><td>10|10-a|
20|20-b|</td><td></td></tr>
<tr><td>T2</td><td>commit;</td><td></td><td></td><td></td></tr>
<tr><td>kv</td><td>10|10-b|
20|20-b|</td><td></td><td></td><td></td></tr>
</table>

Prof. Dr. Ingo Claßen

kv	10|10-a| 20|20-a|	T1	T2
T1	begin transaction isolation level read committed; select * from kv where k=10;	10|10-a|	
T2	begin transaction isolation level serializable; select * from kv order by k;		10|10-a| 20|20-a|
T2	update kv set v = '10-c' where k=10; select * from kv order by k;		10|10-c| 20|20-a|
T1	update kv set v = '10-b' where k=10;	wartet	
T2	commit;		
T1	select * from kv where k=10; commit;	10|10-b|	
kv	10|10-b| 20|20-a|		

kv	10|10-a| 20|20-a|	T1	T2
T1	begin transaction isolation level read committed; select * from kv where k=10 for update;	10|10-a|	
T1			
T2	begin transaction isolation level serializable; select * from kv order by k;		10|10-a| 20|20-a|
T2	update kv set v = '10-c' where k=10;		wartet
T1	update kv set v = '10-b' where k=10;		wartet
T1	commit;		Serialisierungsfehler
T2	rollback;		
kv	10|10-b| 20|20-a|		